

First Named Inventor	James A. Lamb
Serial No.	10/692,939
Filing Date	October 24, 2003
Group Art Unit	2194
Examiner Name	Charles E. Anya
Confirmation No.	2113
Attorney Docket No.	200312021-1
Title: PROGRAM INTERFACE ARCHITECTURE	

**REPLY BRIEF**

**REPLY BRIEF**

Mail Stop: Appeal Brief - Patents  
Commissioner for Patents  
P. O. Box 1450  
Alexandria, VA 22313-1450

**Table of Contents**

I.	Introduction.....	1
II.	Argument .....	2
A.	<i>Examiner's Response to Arguments - Points 1 and 2</i> .....	2
B.	<i>Examiner's Response to Arguments - Point 3</i> .....	2
C.	<i>Examiner's Response to Arguments - Point 4</i> .....	2
D.	<i>Examiner's Response to Arguments - Point 5</i> .....	2
III.	Conclusion .....	6

I. **Introduction**

Appellant filed a Notice of Appeal to the Board of Patent Appeals and Interferences on February 12, 2009. Appellant filed an Appeal Brief on August 12, 2009. An Examiner's Answer addressing Appellant's Appeal Brief was mailed on November 16, 2009. Appellant submits this Reply Brief to address issues raised by the Examiner's Answer.

## II. Argument

### A. Examiner's Response to Arguments - Points 1 and 2

Appellant acknowledges that the objections to claims 17-28 are not reviewable by the Board. Appellant will address the objections upon an indication of allowability of claims on the merits.

### B. Examiner's Response to Arguments - Point 3

Appellant acknowledges that the rejection under 35 U.S.C. § 112, second paragraph is maintained as to claim 17. Appellant contends that it is clear from the record that the scope of claim 17 is reasonably ascertainable to those skilled in the art as evidenced by the Examiner's interpretation of the claim. As such, a rejection of claim 17 as being indefinite is improper. *See MPEP § 2173.05(e)* ("If the scope of a claim would be reasonably ascertainable by those skilled in the art, then the claim is not indefinite.").

### C. Examiner's Response to Arguments - Point 4

Appellant acknowledges that the rejection of claim 23 under 35 U.S.C. § 112, second paragraph is withdrawn. Although the status of claim 29 is not expressly addressed in this regard, Appellant notes that the rejection under 35 U.S.C. § 112, second paragraph as articulated in the Examiner's Answer (page 5, section 2) refers only to claims 17-22. Appellant thus presumes that the rejection of claims 29-32 under 35 U.S.C. § 112, second paragraph is also withdrawn.

### D. Examiner's Response to Arguments - Point 5

While Appellant acknowledges that Bond et al. (U.S. Publication No. 2002/0052727 A1) purports to allow the operation of native and non-native applications on a single platform, Appellant contends that it does so in a wholly different manner than Appellant's claims recite. In this Reply Brief, Appellant will address claim 1, specifically, but the arguments apply to the

remaining independent claims as well. Further support may be found in Appellant's Appeal Brief.

Claim 1 recites, in part, "an operating system layer having executable instructions to provide a first type of operating system" and "wherein the interface module receives program instructions from a program in the application layer written for the second type of operating system and processes the instructions to select either, one of the first number of APIs or one of the second number of APIs." Thus, the program providing the program instructions in claim 1 is a non-native program, as it is written for a type of operating system different than the type of operating system of the operating system layer. As provided in claim 1, in response to the program instructions from this non-native program, the interface module processes the instructions to select an API, and it selects either an API from the first number of APIs, i.e., native APIs, or an API from the second number of APIs, i.e., non-native APIs. Appellant contends that because claim 1 expressly recites "to select either" with respect to the selection of an API, it cannot read on a configuration that provides access to only one set of APIs as there would be no selection to be made between API types if the interface module did not provide access to both native and non-native APIs. Accordingly, a non-native program of Appellant's claims has access to native APIs through the interface module. As noted in Appellant's Appeal Brief, Bond et al. does not teach this implementation. In fact, Bond et al. expressly teaches away from this implementation.

As stated in Bond et al., "The native applications communicate with a set of native APIs 320. . . . The non-native applications communicate with a set of non-native APIs 322." Bond et al., paragraph 0069. Furthermore, Bond et al. goes beyond simply stating that native applications communicate with native APIs whereas non-native applications communicate with non-native APIs. Bond et al. expressly states that the non-native applications cannot call the native APIs. Bond et al., paragraph 0070 ("Without the exemplary kernel emulation, the non-native applications (such as 314a and 314b) would not function. Why? They cannot call the native APIs 320 . . ."). Thus, because Appellant's claims provide non-native programs access to native APIs in direct contrast to Bond et al., a rejection under 35 U.S.C. § 102(e) cannot be supported by the record. Similarly, because Bond et al. expressly teaches away from providing non-native programs access to native APIs, Appellant contends that it also cannot be modified in

a manner necessary to support a rejection under 35 U.S.C. § 103. *See, e.g., In re Ratti*, 270 F.2d 810, 123 USPQ 349 (CCPA 1959) (improper to require a substantial reconstruction and redesign of the elements of the primary reference as well as to change a basic principle under which the primary reference was designed to operate).

The Examiner contends, “This notwithstanding and importantly, the kernel emulator may also provide processor simulators (CPU simulators) for different platforms (page 5 paragraph 0082). This implies that the kernel emulator allows different applications (native and non-native applications) to communicate with plural platforms/operating system by selecting or finding the processor simulator for the target platform/operating system.” Examiner’s Answer, page 23, first full paragraph. Appellant contends that this is a mischaracterization of the Bond et al. reference. In particular, Bond et al. expressly states that native applications are not provided access to the CPU simulators of the kernel emulator, but are instead processed by the native operating system. Bond et al., paragraph 0130 (“At 510 of FIG. 5, an application is loaded (i.e., initialized). At 512, if the application is determined to be a native application, then is processed as normal, at 530, by the OS.”).

The Examiner further states, “When an application (program modules) is loaded-initialized for execution or makes a call (e.g. API call: **NOTE: functionally equivalent to the claimed receiving program instructions for a program in the application layer**), the application type is first determined (i.e. determining whether it is a native or non-native application). If it is a native application, the native program modules APIs is selected and implemented in order to access or communicate with the operating system/kernel (**figure 5 Step 530: NOTE: functionally equivalent to processing the instructions to select the first number of APIs**), however if it is a non-native application, the CPU simulator including a translator (Translator 412) is selected and implemented in order to access or communicate with the operating system/kernel (**figure 5 Step 514: NOTE: functionally equivalent to processing the instructions to select the second number of APIs**).” Examiner’s Answer, page 23, second full paragraph (emphasis in original). However, as noted above, even if Appellant were to accept these assertions as true, which it does not, these assertions do not address the elements of Appellant’s claims. In particular, the program addressed in the claims is a non-native program. Therefore, that Bond et al. selects native APIs for native applications, and selects non-native

APIs for non-native applications, has no bearing on Appellant's claims. As provided in claim 1, in response to the program instructions from a non-native program, i.e., a program in the application layer written for the second type of operating system, the interface module processes the instructions to select an API, and it selects either an API from a first number of APIs for operating on a first type of operating system that is resident on a computing device, i.e., native APIs, or a second number of APIs for operating with a second type of operating system that is not the type of operating system resident on the computing device, i.e., non-native APIs.

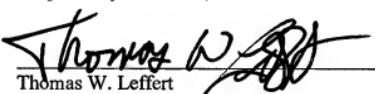
In view of the foregoing, Appellant contends that the rejections under 35 U.S.C. § 102(e) and 35 U.S.C. § 103(a) are unsupported by the record. In particular, Appellant contends that the Bond et al. reference expressly teaches away from a computing device having an interface module coupled between an application layer and an operating system layer providing a first type of operating system, wherein the interface module receives program instructions from a program in the application layer written for a second type of operating system and processes the instructions to select either, one of a first number of APIs for operating on the first type of operating system that is resident on the computing device or one of the second number of APIs for operating with a second type of operating system that is not the type of operating system resident on the computing device. Because each of Appellant's independent claims include the substance of at least these elements, Appellant contends that Bond et al. cannot anticipate any claim subject to appeal. Furthermore, because Bond et al. expressly teaches away from such elements, Appellant contends that Bond et al. also cannot be modified in a manner necessary to support a rejection under 35 U.S.C. § 103(a) of any claim subject to appeal. As such, Appellant contends that every claim that is subject to this appeal is patently distinct from Bond et al., whether taken alone or in combination with other references.

### III. Conclusion

For at least the reasons discussed above, and for reasons as presented in Appellant's Appeal Brief, Appellant submits that the pending claims are patentable. Accordingly, Appellant requests that the Board of Appeals reverse the Examiner's rejections regarding claims 1-11 and 14-32.

Respectfully submitted,

Date: 19 JAN 09



Thomas W. Leffert  
Reg. No. 40,697

Attorneys for Appellant  
HEWLETT-PACKARD COMPANY  
Intellectual Property Administration  
3404 East Harmony Rd.  
Fort Collins, CO 80527-2400